

solutions dispersed throughout the whole data set. For data sets with solutions concentrated around certain portions of the data, the impact of XB-trees is more significant, since many internal nodes can be skipped.

[0103] The present invention provides holistic join algorithms for matching XML query twig patterns, a core operation central to much of XML query processing, both for native XML query processor implementations and for relational XML query processors. In particular, Algorithm TwigStack was shown to be I/O and CPU optimal for a large class of query twig patterns, and practically efficient.

[0104] One skilled in the art will appreciate further features and advantages of the invention based on the above-described embodiments. Accordingly, the invention is not to be limited by what has been particularly shown and described, except as indicated by the appended claims. All publications and references cited herein are expressly incorporated herein by reference in their entirety.

What is claimed is:

1 1. A method of computing an answer to a query path pattern, comprising:
2 (a) associating nodes in a query pattern with respective streams containing positional
3 attributes;
4 (b) associating a respective stack to each node in the query pattern;
5 (c) sorting the stream nodes according to a first one of the positional attributes;
6 (d) determining whether a first set of streams is empty;
7 (e) retrieving a first node in the first set of streams having a minimum value for the
8 first one of the positional attributes;
9 (f) removing partial answers that cannot extend to full answers from a stack
10 associated with the query pattern node;
11 (g) augmenting remaining partial answers with a further stream node;
12 (h) determining whether the further stream node is a leaf node; and
13 (i) producing solutions if the further node is a leaf node.

1 2. The method according to claim 1, wherein the first one of the positional attributes
2 corresponds to left position value.

1 3. The method according to claim 1, wherein the positional attributes include one or
2 more of left position value, right position value, and level number.

1 4. The method according to claim 1, wherein the first set of streams is in a subtree
2 rooted at a given node associated with the query pattern.

1 5. A method of computing an answer to a query twig pattern, comprising:
2 determining a plurality of streams associated with the query twig pattern;
3 determining whether a first set of streams of the plurality of streams is not empty;
4 retrieving a next node in the first set of streams, if not empty, and verifying that
5 the next node has a descendant node in each of the first set of streams associated with the
6 query twig pattern;

7 recursively verifying that the descendant node has a further descendant node in
8 each of the set of streams associated with the query twig pattern;
9 cleaning a stack containing partial solutions corresponding to a parent node of the
10 next node if the next node is not a root node,
11 determining, if the next node is a root node and after cleaning the stack, whether
12 the next node is a root node or a stack associated with a parent node of the next node is
13 not empty;
14 cleaning the stack associated with the next node if the next node is a root node or
15 the stack associated with the parent node of the next node is not empty, otherwise
16 advancing the stream containing the next node;
17 after cleaning the stack associated with the next node, adding the next node to the
18 stack containing partial solutions; and
19 advancing the stream containing the next node if the next node is not a leaf node,
20 otherwise, generating a solution to the query twig pattern.

1 6. The method according to claim 5, further including generation a solution with
2 blocking.

1 7. The method according to claim 5, further including determining positional attributes
2 of nodes in a document.

1 8. The method according to claim 7, wherein the positional attributes includes one or
2 more of document ID, left position, right position, and level number.

1 9. The method according to claim 8, further including using one or more indexes for the
2 nodes.

1 10. An article comprising:
2 a storage medium having stored thereon instructions that when executed by a
3 machine result in the following:

- 4 (a) associating nodes in a query pattern with respective streams containing positional
- 5 attributes;
- 6 (b) associating a respective stack to each node in the query pattern;
- 7 (c) sorting the stream nodes according to a first one of the positional attributes;
- 8 (d) determining whether a first set of streams is empty;
- 9 (e) retrieving a first node in the first set of streams having a minimum value for the
- 10 first one of the positional attributes;
- 11 (f) removing partial answers that cannot extend to full answers from a stack
- 12 associated with the query pattern node;
- 13 (g) augmenting remaining partial answers with a further stream node;
- 14 (h) determining whether the further stream node is a leaf node; and
- 15 (i) producing solutions if the further node is a leaf node.

- 1 11. An article, comprising:
- 2 a storage medium having stored thereon instructions that when executed by a
- 3 machine result in the following:
- 4 determining a first set of streams associated with the query twig pattern;
- 5 determining whether the first set of streams is not empty;
- 6 retrieving a next node in the first set of streams, if not empty, and verifying that
- 7 the next node has a descendant node in each of the streams associated with the query twig
- 8 pattern;
- 9 recursively verifying that the descendant node has a further descendant node in
- 10 each of the plurality of streams associated with the query twig pattern;
- 11 cleaning a stack containing partial solutions corresponding to a parent node of the
- 12 next node if the next node is not a root node,
- 13 determining, if the next node is a root node and after cleaning the stack, whether
- 14 the next node is a root node or a stack associated with a parent node of the next node is
- 15 not empty;

16 cleaning the stack associated with the next node if the next node is a root node or
17 the stack associated with the parent node of the next node is not empty, otherwise
18 advancing the stream containing the next node;
19 after cleaning the stack associated with the next node, adding the next node to the
20 stack containing partial solutions; and
21 advancing the stream containing the next node if the next node is not a leaf node,
22 otherwise, generating a solution to the query twig pattern.

1 12. A computer, comprising:

2 a memory having stored instructions that when executed result in the following:
3 (a) associating nodes in a query pattern with respective streams containing positional
4 attributes;
5 (b) associating a respective stack to each node in the query pattern;
6 (c) sorting the stream nodes according to a first one of the positional attributes;
7 (d) determining whether a first set of streams is empty;
8 (e) retrieving a first node in the first set of streams having a minimum value for the
9 first one of the positional attributes;
10 (f) removing partial answers that cannot extend to full answers from a stack
11 associated with the query pattern node;
12 (g) augmenting remaining partial answers with a further stream node;
13 (h) determining whether the further stream node is a leaf node; and
14 (i) producing solutions if the further node is a leaf node.

15

1 13. A computer, comprising:

2 a memory having stored instructions that when executed result in the following:
3 determining a plurality of streams associated with the query twig pattern;
4 determining whether a first set of streams of the plurality of streams is not empty;
5 retrieving a next node in the first set of streams, if not empty, and verifying that
6 the next node has a descendant node in each of the plurality of streams associated with
7 the query twig pattern;

8 recursively verifying that the descendant node has a further descendant node in
9 each of the plurality of streams associated with the query twig pattern;
10 cleaning a stack containing partial solutions corresponding to a parent node of the
11 next node if the next node is not a root node,
12 determining, if the next node is a root node and after cleaning the stack, whether
13 the next node is a root node or a stack associated with a parent node of the next node is
14 not empty;
15 cleaning the stack associated with the next node if the next node is a root node or
16 the stack associated with the parent node of the next node is not empty, otherwise
17 advancing the stream containing the next node;
18 after cleaning the stack associated with the next node, adding the next node to the
19 stack containing partial solutions; and
20 advancing the stream containing the next node if the next node is not a leaf node,
21 otherwise, generating a solution to the query twig pattern.